In another embodiment, each indicator deployment agent is managed either by the agent machine that manages the configuration agent associated with the indicator deployment agent, or by a different agent machine.

The invention, along with its characteristics and advantages, will emerge more clearly through the reading of the description given in reference to the attached drawings, in which:

- Fig. 1 represents a simplified schema of a computer system in which the method of the invention can be applied.

- Fig. 2 represents the process for deploying a monitoring method.

Setting up a monitoring of a computer system requires the configuration, then the deployment of this monitoring. A configuration of a monitoring is defined by a set of domains and indicators. A domain symbolically contains a set of equipment units of the computer system to be monitored. An indicator designates both the value characterizing the status or the operation of a set of equipment to be monitored, and the equation that makes it possible to calculate this value.

Configuring a monitoring consists of specifying for each indicator the domain or domains for which the indicator could be deployed. Deploying an indicator for a given domain consists of instantiating this indicator for each unit of equipment belonging to this domain. The method and the device according to the invention make it possible to perform this deployment.

The invention will now be described in reference to Figs.1 and 2.

As explained above, a computer system (1) comprises at least one local area network (10, 20) that communicates with a central system (2) or manager through a wide area network (3). Each local area network (10, 20) comprises at least one unit of computer equipment (101, 102, 201, 202) called a resource.

In the prior art, the monitoring of all the resources (101, 102, 201, 202) is handled by means of remote agents (SNMP or CMIP agents, depending on the management protocol in question) installed in each resource (101, 102, 201, 202). These agents allow a manager to collect information on the resources (101, 102, 201, 202) by measuring given parameters, for example by sending an SNMP request GetRequest to the SNMP agent in question, then to transmit the collected information to the manager (2) by means of a request, for example GetResponse in the SNMP world. The manager (2) then evaluates the indicators from this received information. It is understood that all the requests sent by the manager (2) in order to gather monitoring information and the responses to these requests from the agents produces

considerable congestion in the wide area network (3), consequently diminishing its performance.

According to the invention, all or part of the evaluation of the indicators is distributed at the local area network level (10, 20) to indicator agents.

To do this, for each indicator (In) characterizing a particular operation or status of a resource (101, 102, 201, 202), a specific indicator agent (A1, A2, B1, B2) is installed in the resource (101, 102, 201, 202). In the exemplary embodiment represented in Fig. 1, each agent (A1, A2, B1, B2) is responsible for evaluating a different indicator (In). In other words, each indicator (In) makes it possible to determine a different operation or a different status in each resource (101, 102, 201, 202). However, each agent (A1, A2, B1, B2) described in reference to Fig. 1 can be present, for example, in all of the resources (101, 102, 201, 202) or all the resources of the same type in the computer system (1).

For example, a first agent (A1) can evaluate an indicator (IA1) whose formula makes it possible to know, for example, the number of alarms per minute emitted by a first resource (101). It is understood that this agent (A1) can be installed in all the resources (101, 102, 201, 202) of a computer system (1).

The indicator agents (A1, A2, B1, B2) are elementary programming and execution entities. The indicator agents (A1, A2, B1, B2) are autonomous in order to give the structure the property of scalability by making it easy to add or delete one or several indicator agents in the architecture. They are made autonomous by correctly defining the interface for these agents. They are also autonomous in terms of communication. They communicate using notifications and not procedure calls, which makes it possible to open up the architecture. This communication is asynchronous, thereby making it possible to obtain a better parallelization of the indicator agents and hence a better scalability of the architecture.

The indicator agents (A1, A2, B1, B2) can be persistent objects, and their status is derived from data that, in this case, can exist outside any execution structure and can be stored in persistent E^2PROM or hard disk memories. An indicator agent (A1, A2, B1, B2) is an independent machine that reacts to notifications.

Each agent (A1, A2, B1, B2) is an object of a class and is identified by a respective identifier id(A1), id(A2), id(B1), id(B2) that is unique in the computer system (1) in order for communications between indicator agents to be possible, and that makes it possible to locate them. An agent can be created in a remote server. The creation entity is responsible for creating the identifier. The structure of the identifier must take into account the static location

of the agents in order to allow the system to forward the notifications, and to allow local "production" of the identifiers for the remote agents. The identifier of an agent comprises the following three parts:

- the identification (idsac) of the agent server hosting the creation agent,
- the identification (idsal) of the agent server hosting the agent created, and
- a postmark (el), local to the agent server hosting the creation agent.

A "factory" agent is present in all of the agent servers and is responsible for creating objects in the server It makes it possible to manage the creation of remote agents.

According to the invention, each indicator agent (A1, A2, B1, B2) is responsible for evaluating one indicator (In). In order to do this, it must comprise properties that make it possible to collect a measurement (Mn) on the resource (101, 102, 201, 202) with which it is associated, and communication properties in order to request or receive, as necessary, at least one other indicator evaluated by another indicator agent (A1, A2, B1, B2) according to the invention, for example by means of notifications. These properties are obtained by means of specific program modules constituting each indicator agent (A1, A2, B1, B2). Thus, an indicator agent (A1, A2, B1, B2) comprises a collection module and/or a communication module. Likewise, an indicator agent (A1, A2, B1, B2) also comprises an evaluation module for calculating the indicator itself from the formula, using the information collected by the other two modules. In this evaluation module, the equation that defines the value of an indicator (IA1 for example) can refer to a possibly empty set of other indicators (IA2, IB1, for example).

The indicator agents use, for example, a communication protocol that is preferably asynchronous, which means that when an indicator agent sends a request for collecting a measurement on a resource of the system or a notification to another indicator agent, it does not wait for the response to this request or notification before possibly sending a new request or notification. In other words, the sending of requests and notifications is performed in parallel. Likewise, the management protocol can be asynchronous. For example, the asynchronous management protocol used is the protocol SNMP or the protocol CMIP.

However, it is possible to use a synchronous protocol, but the performance in that case is inferior, particularly in terms of speed in evaluating the indicators. Moreover, it is appropriate to install an asynchronous device for collecting and storing measurements so that the indicator agent that sent the request need only consult the memory of this measurement